See Attachment A on CD-Rom Disc

A

B

Computer Modeling of the Local Telephone Network

C. A. Bush, D. M. Kennet, J. Prisbrey and W. W. Sharkey
Federal Communications Commission

and

Vaikunth Gupta
Panum Telecom, LLC

October 1999

Computer Modeling of the Local Telephone Network

1. Introduction

Engineering process models have been developed in recent years as an alternative to more traditional econometric and accounting approaches to cost measurement. Because econometric models rely on assumptions of (smooth) functional forms, engineering process models offer a more detailed view of cost structures than is possible using econometric data. In addition, engineering models are better suited for modeling forward-looking (long run) costs since they rely much less on historical data than econometric models. In the regulatory arena, cost models have also been actively promoted as potential tools for determining the cost of universal service support in high cost areas, and potentially for determining the forward-looking cost of network elements which could be used in other regulatory situations.

The model we present in this paper was originally developed as an alternative to models being promoted for the purposes mentioned earlier by industry players in a variety of regulatory settings. The model combines appropriate principles of engineering design for the local loop, switching and interoffice networks with economic principles of cost minimization. The result draws on expertise in several disciplines and, also because it draws freely from engineering principles displayed in other models, is called the Hybrid Cost Proxy Model, or HCPM.

The HCPM currently consists of two independent modules – a customer location module and a loop design module. The customer location module first groups individual geographic locations of telephone customers into clusters, based on engineering considerations. Next, the customer location module determines a grid and microgrid overlay for each cluster, and places each customer location into the correct microgrid cell. The loop design module determines the total investment required for an optimal distribution and feeder network by building loop plant to the designated customer locations represented by populated microgrid cells. When used with a source of geocoded customer locations and a maximum copper reach of 18,000 feet, a uniform microgrid size of 360 feet can be maintained. All customer locations can therefore be determined with an error of not more than several hundred feet.

All HCPM modules are written in high level programming languages, and compiled versions can be supported on a number of computing environments. In the following sections, a more detailed, and more formal description of each of the modules will be presented. In section 2, we review the regulatory environment in which proxy cost models have been developed and evaluated by regulatory agencies. Section 3 provides an overview of the fundamental algorithms and modeling techniques used in the HCPM. Section 4 compares HCPM with two of the alternative industry sponsored models. Appendix A describes how specific cost estimates are developed based on the model algorithms combined with tables which reflect both input prices and engineering constraints. Appendix B describes data input requirements in some additional detail.

2. A Review of the FCC's Decision to Adopt a Proxy Model

In 1996 the U.S. Congress passed the Telecommunications Act[1] which sought to establish "a pro-competitive, de-regulatory national policy framework" for the United States telecommunications industry. In the two full years following this act, the Federal Communications Commission (FCC) has undertaken proceedings on universal service,[2] interstate access charge reform,[3] and local exchange competition[4] to

[1] 47 U.S.C. §§ 151 *et. seq.*
[2] *In the Matter of Federal-State Joint Board on Universal Service*, CC Docket No. 96-45

overhaul its current regulations in light of the 1996 Act. In these proceedings, the Commission has examined, in varying degrees, the use of forward-looking economic cost methodologies as a basis for determining universal service support levels, cost-based access charges, and pricing for interconnection and unbundled network elements. The 1996 Act has fundamentally changed telecommunications regulation by replacing the framework of government-recognized monopolies with one in which federal and state governments work in tandem to promote efficient competition and to remove entry barriers and regulations that protect monopolies. The 1996 Act, when fully implemented, may greatly reduce the legal, regulatory, economic, and operational barriers to entry in the local exchange and exchange access market, while preserving and advancing enhanced universal service goals.

The local competition provisions of the Act confer three fundamental rights on potential competitors to incumbent local exchange carriers (LECs): the right to interconnect with other carriers' networks at rates based on cost; the right to obtain unbundled network elements at cost-based rates; and the right to obtain an incumbent LEC's retail services at wholesale discounts in order to resell those services.[5] The Act also requires a fundamental restructuring of current regulatory mechanisms for funding universal service goals. Prior to the 1996 Act, several explicit interstate universal service programs provided assistance to small incumbent local exchange carriers and carriers that served rural and high-cost areas. Other mechanisms also have historically contributed to maintaining affordable rates in rural areas, including subsidies implicit in geographic toll rate averaging, intrastate rates, and interstate access charges. Section 254 of the 1996 Act directed the FCC to reform universal service support mechanisms to ensure that they are compatible with the goals of the 1996 Act. After receiving the recommendations of a Federal-State Joint Board, the Commission adopted a Universal Service Order in May 1997. In the Universal Service Order, the Commission adopted a forward-looking economic cost methodology to calculate support for non-rural carriers. Under this methodology, a forward-looking economic cost mechanism would be used to estimate non-rural carriers' forward- looking economic cost of providing services in high-cost areas.

Forward-looking economic computer based cost models can enable regulatory authorities to estimate the forward-looking cost of network facilities and services without having to rely on detailed cost studies, prepared by incumbent local exchange carriers, that otherwise would be necessary. In addition, a publicly available cost proxy model can be useful to regulators by providing an independent check on the accuracy of incumbent LEC cost studies. During the course of the model development process, several industry sponsored models were submitted to the FCC for evaluation. These include the Benchmark Cost Proxy Model ("BCPM") sponsored by US West, Sprint and Bell South, and the HAI model sponsored by AT&T and MCI. Simultaneously, staff members of the FCC worked on an internal model known as the Hybrid Cost Proxy Model ("HCPM") which incorporated elements of both of the industry models in addition to a set of new loop design and clustering algorithms developed internally.

In order to implement a forward-looking approach to costing of the local exchange telephone network using a computer model, the FCC, in its universal service proceedings, established a multi-phase plan that divided questions related to the cost models into "platform design" issues and "input value" issues. In October 1998 the Commission adopted a synthesis model consisting of the HCPM clustering and loop design modules in combination with HAI switching, transport and expense modules. In May 1999, the Commission released the Inputs Further Notice in which it proposed and sought comment on a complete set of input values for use in the model, such as the cost of switches, cables, and other network components. In October 1999 the Commission adopted a final inputs order which adopted a set of inputs for the model

---

[3] *In the Matter of Access Charge Reform*, Notice of Proposed Rulemaking, FCC No. 96-488, CC Docket No. 96-262 (rel. Dec. 24, 1996).

[4] *Implementation of the Local Competition Provisions of the Telecommunications Act of 1996*, CC Docket No. 96-98, FCC 96-325 (released August 8, 1996), Order on Reconsideration *Implementation of the Local Competition Provisions of the Telecommunications Act of 1996*, CC Docket No. 96-98, 11 FCC Rcd 13042 (1996), *petition for review pending sub nom. and partial stay granted, Iowa Utilities Board v. FCC*, No. 96-3321 and consolidated cases (8th Cir., Oct. 15, 1996), *partial stay lifted in part, Iowa Utilities Board et. al v. FCC*, No. 96-3321 and consolidated cases (8th Cir., Nov.1, 1996).

[5] 47 U.S.C. §§ 251(c)(2)-(4), 252(d)(1).

along with minor modifications to the platform. In a separate order an explicit methodology for determining non-rural carriers' support to begin on January 1, 2000 was adopted.

3. Overview of the HCPM Modules

The HCPM differs from previous models proposed by industry groups in two important respects. Since the model is written in a high level compiled programming language, it is not constrained by the memory and storage limitations of the spreadsheet program (Microsoft Excel) which performed the basic computations in all previous models. The cluster algorithm, which performs the first level of network design by grouping customers into serving areas, is an explicit component of the model which is open to inspection and review by all parties. While the HAI model also used a similar clustering procedure, this model required the clustering to be done in a pre-processing stage, so that only the outputs of the clustering process were publicly available. In addition, the design limitations of Excel meant that the exact customer locations of customers in serving areas determined by the clustering process were significantly distorted by aggregation into stylized rectangles each containing a uniform density of customers. As a result, it is possible for the HCPM to explicitly design loop plant to reach individual customer locations as determined from geocoded location data.

The second significant difference between HCPM and earlier models is its explicit use of cost minimization methods in many aspects of loop plant design. Most significantly, the model uses an algorithm developed for network planning purposes in both its feeder and distribution segments. This algorithm selects a feeder or distribution routing network by weighing the benefits of minimizing total route distance (and therefore structure costs) and minimizing total cable distance (and therefore cable investment and maintenance costs.) The model attempts to optimize the trade-off between distribution plant, feeder plant and loop electronics by considering alternative configurations of distribution plant in which each serving area is subdivided into two parts. In the divided serving area, customers can be located closer to a serving area interface, but at a cost of additional feeder plant and loop electronics.

In the synthesis model adopted by the FCC in its universal service order, the HAI switching, interoffice and expense modules were also adopted. Switching costs are estimated by using host-remote data from the Local Exchange Routing Guide (LERG). Small capacity and limited function remote switches are connected to full function host switches through SONET ring connections. The interoffice module interconnects host and tandem switches using additional SONET rings. An expense module converts investments into an estimate of the monthly cost of providing service. Expenses include: plant-specific expenses, such as maintenance of facilities and equipment expenses; plant non-specific expenses, such as engineering, network operations, and power expenses; customer service expenses, such as marketing, billing, and directory listing expenses; and corporate operations expenses, such as administration, human resources, legal, and accounting expenses.

3.1 The Customer Location Module

The "synthesis" model adopted in the FCC's Platform Order allows the user to estimate the cost of building a telephone network to serve subscribers in their actual geographic locations, to the extent these locations are known. To the extent that the actual geographic locations of customers are not available, the Commission determined that the synthesis model should assume that customers are located along roads. The model's estimate of the cost of serving the customers located within a given wire center's boundaries includes the calculation of switch size, the lengths, gauge, and number of copper and fiber cables, and the number of DLCs required. These factors depend, in turn, on how many customers the wire center serves, where the customers are located within the wire center boundaries, and how they are distributed within neighborhoods. Particularly in rural areas, some customers may not be located in neighborhoods at all but, instead, may be scattered throughout outlying areas.

Once the customer locations have been determined, the model employs a clustering algorithm to group customers into serving areas in an efficient manner that takes into consideration relevant engineering constraints. In this section we describe a method of modeling customer location based on cluster analysis.

This approach, implemented in the module CLUSTER, is designed to accept as input a set of geocoded locations for every customer. It then performs a "rasterization" procedure which assigns customers to small microgrid cells.[6] A cluster algorithm then groups the set of raster cells into natural clusters. Finally, a square grid is constructed on top of every cluster, and all customer locations in the cluster are assigned to a microgrid cell for further processing by the loop design module. The cluster module can also accept Census block level data as an input. In this case every block which is larger than a raster cell is broken up into smaller blocks, each no larger than a raster cell, and the population of the original block is distributed uniformly among the new blocks.

3.1.1 The Clustering Algorithm

One of the primary tasks faced by the HCPM is to identify clusters of customer locations. Each customer in a particular cluster, or serving area, will then be connected to the feeder system through a single interface, the Serving Area Interface or SAI.

The clustering task is difficult because both engineering constraints and the general pattern of customer locations must be considered. There are two main engineering constraints. First, a serving area is limited to a certain number of lines by the capacity of the SAI. Second, a serving area is limited to certain geographic dimensions by current technology since as distance increases beyond a critical value, service quality is degraded.

Given the engineering constraints, one could create feasible serving areas by simply placing a grid containing cells of an appropriate dimension over the entire wirecenter. For this to be a cost-effective approach, however, customers would have to be located in a relatively uniform pattern across the entire wirecenter. In actuality, people tend to live clumped together in towns and communities. Moreover, the distribution of population in large cities, small towns and rural regions differs significantly among states in the U.S. and among countries. Under these conditions, a gridding approach applied to non-uniform population data may divide a natural grouping of customers into different serving areas when a single serving area would be more cost effective.

The objective of a clustering algorithm is to create the proper number of feasible serving areas. A clustering algorithm must consider both fixed and variable costs associated with each additional serving area. A fixed cost gives a clear incentive to create a small number of large clusters, rather than a larger number of smaller-clusters. On the other hand, with fewer clusters the average distance of a customer from a central point of a cluster, and consequently the variable costs associated with cable and structures, will be larger. In moderate to high density areas, it is not clear, *a priori*, what number of clusters will embody an optimal trade-off between these fixed and variable costs. However, in low density rural areas, it is likely that fixed costs will be the most significant cost driver. Consequently, a clustering algorithm that generates the smallest number of clusters could be expected to perform well in rural areas.

While statisticians have studied a wide variety of clustering algorithms,[7] there are two basic approaches to clustering: the agglomerative or bottom-up approach, and the divisive or top-down approach. Each approach starts with an initial state where each customer location belongs to a particular cluster. The initial state is then improved upon according to some rule, until no more improvements can be made. The clustering module for the HCPM contains three alternative algorithms which represent implementations of both of the above approaches. The default method, a divisive algorithm, has substantial advantages over both the agglomerative algorithms, because it tends to create the smallest number of clusters and is also by far the most efficient algorithm in terms of computer run-time.

---

[6] The size of a microgrid cell is specified by the user. The recommended size is a square with side equal to 150 feet, but smaller cells can be chosen at the expense of increased computing time.
[7] For information about different clustering methods see: Brian S. Everitt, 1993, *Cluster Analysis, Third Edition*, London: Arnold.

In the initial state for the default divisive approach, each location belongs to a single parent cluster. This initial state is improved upon by dividing the parent cluster into a new parent cluster and a child cluster. This step increases the total number of clusters by one. The improvement step is repeated until every cluster is feasible from an engineering standpoint.[8] A child cluster is created from the parent cluster by choosing the customer location furthest from the parent's line-weighted center as an initial child cluster member. Then, customer locations that are closer to the center of the child cluster than they are to the center of the parent cluster are reassigned in an iterative manner, recalculating the cluster centers at each step. Customer locations are added to the child cluster until it is full, *i.e.* until no more locations can be added without violating engineering constraints.

Alternatively, in the initial state for all agglomerative approaches, each location belongs to its own unique cluster. This initial state is improved upon by merging the two closest clusters together, reducing the total number of clusters by one. The improvement step is repeated until merging is no longer feasible from an engineering standpoint. The HCPM cluster module contains two agglomerative algorithms that differ only in how they measure the distance between clusters. In the standard agglomerative algorithm, distance is measured from the line-weighted center of one cluster to the line-weighted center of another. In the nearest-neighbor algorithm, distance is measured from the two customer locations, one in each cluster, that are closest together. The nearest-neighbor method contains an additional constraint, used in the HAI model, that no customer locations are joined if the distance between them is more than two miles.

Once one of the clustering algorithms has been run, experience has demonstrated that the initial result can generally be improved by reassigning certain customer locations to different clusters. The cluster module contains two optimization routines that perform these reassignments. The first routine, called "simple reassignment", reassigns a customer location to a different cluster, if the location is closer to that cluster's center. The routine operates sequentially, taking account of both the maximum distance and line count constraints. After the reassignment, cluster centers are re-computed and the routine is repeated.[9] The process continues until no more reassignments can be made. The second routine, called "full optimization", considers customer locations one-by-one. It measures the effect each customer location has on the location of cluster centers, and moves a location from one cluster to another if the total distance from all customer locations to their cluster centers is reduced. The routine moves the customer location that gives the most distance reduction at each step. It continues until no more distance reduction is possible.

Each time a cluster center is moved, or the number of lines in a cluster is reduced, new reassignments may become possible that were not feasible before. While simple reassignment takes very little computer time, full optimization can be a slow process if the number of customer locations is large. We have found that an efficient, but effective, method of optimization routine consists of three steps: simple assignment, followed by full optimization, and completed by a final round of simple reassignment. If the number of customer locations is large, the intermediate full optimization step is eliminated. In some cases, these optimization methods generate very large improvements. The total distance from customer locations to their clusters' centers is sometimes reduced by as much as 50-60%. More typically, it is reduced by 10-30%.

As a final step, the cluster module computes potential locations for either one SAI, or for a pair of SAIs. The location for a single SAI is simply the line-weighted center of the cluster. The locations for a pair of SAIs are determined by dividing each cluster into a parent and child. The module then reports the line-weighted centers of the parent and child as potential locations of a pair of SAIs. The actual number of SAIs used is determined within the loop design module.

To improve computation speed, the cluster module uses raster cells to store individual customer location data. A raster is a grid covering the entire wire center. One point in each cell is used to represent all the customer locations that fall into that cell. We use the line-weighted average of all customer locations that fall within the raster cell as the raster point. If there is only one location in the raster cell, the

[8] This and other stopping rules in the divisive algorithm can be adjusted to increase performance.

[9] Note that the cluster's center may or may not be the line-weighted center after this step. The cluster's center is only changed if no engineering constraints are violated. Otherwise, the center is not moved.

raster point is the exact customer location. The clustering algorithms can then operate on raster points rather than actual customer locations. By choosing the size of the raster cell, one can either increase or decrease the number of points the clustering algorithm considers. For maximum speed, one should pick a large raster size, which reduces the number of points the algorithm must consider. On the other hand, for maximum accuracy one should pick a very small raster size. The model currently suggests a default raster size of 150 feet, which we believe represents an acceptable compromise between run time and accuracy.

Within the cluster module, the user can also define the engineering constraints by picking the distance limit, which is the maximum distance a customer location can be from the cluster center, and by picking the line limit, which is the maximum line capacity of an SAI. To allow optimal flexibility to the optimization routines, the user can also pick the line fill percent. The line fill percent specifies the percentage of the line limit that will be used as a constraint by the initial divisive clustering algorithm. Since the full line constraint is not met initially, the optimization routines can actually reassign locations from cluster to cluster. This consideration is important only in areas with high population density. In most rural areas, the line constraint is never met.

Output from the cluster module is illustrated in Figure 1 below. Each color-coded cluster is represented by a set of customer locations connected to the cluster center by a straight line. Each cluster represents a single feasible serving area.

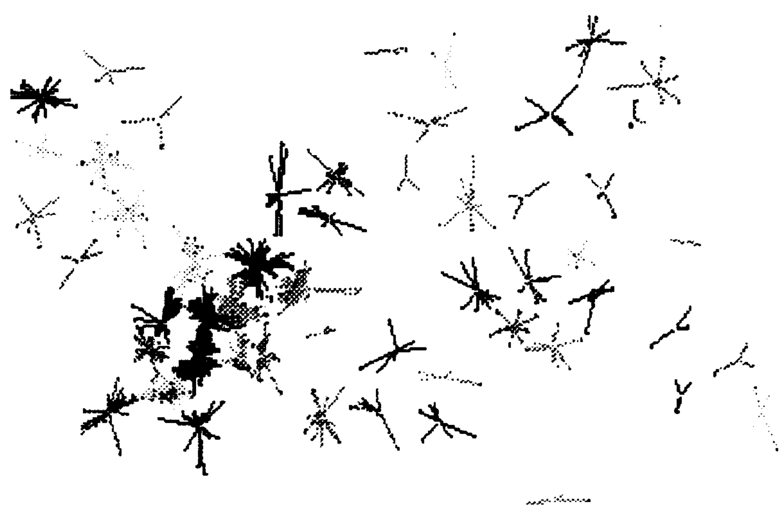


Figure 1: Clusters for the Wire Center GNSNCOMA

3.1.2 Defining a Grid Overlay for each Cluster

The cluster module reports as output a set of one or more clusters for each wire center. For each cluster the following information is provided:

- the coordinates of each customer location
- the number of business and residential lines associated with each location

- terrain data for the cluster, consisting of bedrock depth, rock hardness, soil type, depth of water table, minimum slope, and maximum slope

The final step in the customer location module is to convert the above data into a form that can be utilized by the loop design module. For wire centers that are sufficiently small, it would be possible to build plant to the exact customer locations determined by the geocoded data as processed by the cluster module. For larger wire centers, which may have 20,000 - 100,000 or more individual customer locations, it would be extremely time consuming to build distribution plant directly to each individual location. As in the clustering module, an acceptable compromise between absolute accuracy and reasonable computing time can be achieved by defining a grid on top of every cluster and assigning individual customer locations to microgrid cells within the grid. Customers within each microgrid cell are assumed to be uniformly distributed within the cell. If multiple customer locations fall in a single microgrid cell, the cell is divided into lots as explained in the following section. Loop plant can therefore be designed specifically to reach only populated microgrid cells, and the individual customer lots within each microgrid. For large wire centers, the number of populated microgrid cells will be much less than the number of customer locations, even when a relatively small microgrid size is specified. Therefore, with this approach it is possible to place an upper bound on computing time, while simultaneously placing a bound on the maximum possible error in locating any individual customer.

The process described above is accomplished by the program CLUSINTF. This program takes the outputs of the cluster module and converts them into binary files that are recognized by the loop design module FEEDDIST. CLUSINTF first constructs a square grid around the customer locations within each cluster.[10] Next it constructs a matrix of micrograms. If the user specifies a target microgrid size, the algorithm divides the grid into the number of squares that most closely approximates the desired size.[11] If no size is predetermined, then by default the algorithm creates a 50 by 50 matrix of cells in each grid.[12] After assigning customers to microgrids, CLUSINTF computes the convex hull of the set of all customer locations in each cluster and computes the area of this convex hull. Given the line counts reported by CLUSTER, it is then possible to compute the customer density of each cluster. FEEDDIST uses these densities to compute the cost of installing distribution plant.

3.2. Loop Design Algorithms

A telephone network must allow any customer to connect to any other customer. In order to accomplish this task, a telephone network must connect customer premises to a switching facility (wirecenter), ensure that adequate capacity exists in that switching facility to process all customers' calls that are expected to be made at peak periods, and then interconnect that switching facility with other switching facilities to route calls to their destinations. Within the boundaries of each wire center, the wires and other equipment that connect the central office to the customers' premises are known as outside plant. Outside plant can consist of either copper cable or a combination of optical fiber and copper cable, as well as associated electronic equipment. Copper cable generally carries an analog signal that is compatible with most customers' telephone equipment. The range of an analog signal over copper is limited, however, so thicker, more expensive cables or loading coils must be used to carry signals over greater distances. Optical fiber cable carries a digital signal that is incompatible with most customers' telephone equipment, but the quality of a signal carried on optical fiber cable is superior at greater distances when compared to a signal carried on copper wire. Generally, when a neighborhood is located too far from the wire center to be served by copper cables alone, an optical fiber cable will be deployed to a point within the neighborhood,

[10] The lower left corner of the grid is determined by the minimum of the x and y coordinates over each of the customer locations, minus a small amount to ensure that no customer location is placed on the boundary of the grid. A similar adjustment is made for the upper right corner of the grid.
[11] FEEDDIST has been designed to accept a maximum of 2500 microgrid cells for any individual cluster. The recommended default for the maximum distance of any customer from the SAI is 18,000 feet. If a square grid which is 18 kilofeet on a side is divided into 2500 microgrid cells, each microgrid will be a square of 360 feet on a side. This is the recommended default size of a microgrid.
[12] If microgrids are generated automatically, then each grid will have a different size of microgrid, since the dimensions of the grid are determined by the dimensions of the cluster.

where a piece of electronics equipment will be placed that converts the digital light signal carried on optical fiber cable to an analog, electrical signal that is compatible with customers' telephones. This equipment is known as a digital loop carrier remote terminal, or DLC, which is connected to a serving area interface (SAI). From the SAI, copper cables of varying gauge extend to all of the customer premises in the neighborhood. Where the neighborhood is close enough to the wire center to be served entirely on copper cables, copper trunks connect the wire center to the SAI, and copper cables will then connect the SAI to the customers in the serving area. The portion of the loop plant that connects the central office with the SAI or DLC is known as the feeder plant, and the portion that runs from the DLC or SAI throughout the neighborhood is known as the distribution plant.

Given the inputs from the customer location module, the logic of the loop design module is straightforward. Within every microgrid with non-zero population, customers are assumed to be uniformly distributed. Each populated microgrid is divided into a sufficient number of equal sized lots, and distribution cable is placed to connect every lot. These populated microgrids are then connected to the nearest concentration point, called a serving area interface (SAI), by further distribution plant. During this phase of the loop design algorithms, the heterogeneity of microgrid populations, and the locations of populated microgrids are explicitly accounted for. Finally, the SAIs are connected to the central office by feeder cable. On every link of the feeder and distribution network, the number of copper or fiber lines and the corresponding number of cables are explicitly computed. The total cost of the loop plant is the sum of the costs incurred on every link.

Distribution consists of all outside plant between a customer location and the nearest serving area interface (SAI). Distribution plant consists of backbone and branching cable where branching cable is closer to the customer location. Feeder consists of all outside plant connecting the central office main distribution frame, or fiber distribution frame, to each of the SAIs. Feeder cable consists of main feeder, sub-feeder and sub-sub-feeder routes.

Technology choices for both feeder and distribution plant are constrained by user specified threshold values for the distance from the central office to the most distant customer served from a given terminal. The model recognizes the following distance thresholds.[13]

Copper gauge crossover: the maximum distance served by 26 gauge analog copper
Copper distance threshold: the maximum distance served by 24 gauge copper
Copper-T1 crossover: the maximum feeder distance served by analog copper feeder cable
T1 fiber crossover: the maximum feeder distance served by T1 technology[14]

The model assumes the following progression of technologies as distance increases: 26-gauge copper, 24 gauge copper, T1 on copper, and fiber.[15] The model treats these thresholds as non-binding constraints in a cost optimization. For example, if relative prices indicate that fiber is less expensive than T1, fiber will be chosen even if the feeder distance is less than the T1-fiber crossover. HCPM explicitly computes the cost of the feeder plant which connects each terminal to the central office, and, subject to the distance constraints described above, selects the cost minimizing technology.

3.2.1 Distribution Plant Design

The distribution portion of the loop design module determines the cost of distribution plant for each cluster in isolation (ignoring information from all neighboring clusters). The algorithms described in

[13] The first two thresholds refer to distances from the central office to the furthest customer served in a particular grid. The second two thresholds refer to distance from the central office to an SAI.
[14] With traditional T1 technologies using repeaters, there is no maximum distance constraint, and the crossover point is determined by economic considerations. With HDSL technologies, both the maximum distance constraint and the economic crossover point must be accounted for. We note that by setting the T1-fiber crossover equal to the copper-T1 crossover, the user can instruct the model to ignore T1 technologies in the feeder network.
[15] That is, copper gauge crossover ≤ copper T1 crossover ≤ T1 fiber crossover.

the following sections compute the cost of all plant that is required to connect each customer within the cluster to the nearest SAI.

Distribution Plant Within a Microgrid

Each microgrid is divided into lots based on microgrid population. Distribution cable is built to touch every lot in the cell, as illustrated in Figure 2. Backbone cables, which connect cells to the SAI, are assumed to run horizontally and branching cables within a cell are assumed to run vertically.

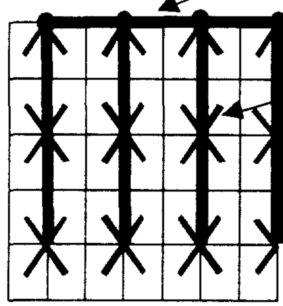
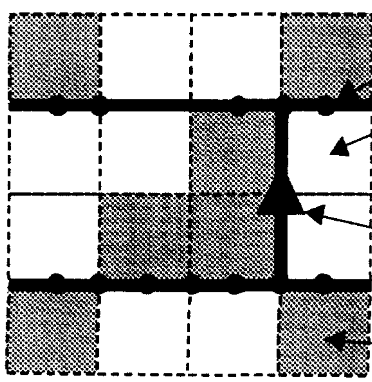


Figure 2: Distribution Cable Within a Cell

Branching cable is assumed to follow every other vertical lot boundary, beginning at the upper right hand corner of the lower-left most lot as shown in Figure 2. Drop cable is designed to serve groups of four properties whenever possible. If drop wire runs to the center of a lot, its length is equal to one half of the diagonal of the rectangle which defines the lot. This represents the maximum possible drop length in a lot. Alternatively, if the residence is assumed to be located on the midpoint of the lot frontage, the drop length would be equal to one half of the width of a lot. HCPM assumes that actual drop length is a convex combination of these two extreme possibilities, weighted by a user parameter λ (set by default equal to ½).

Connection of Microgrids to the Nearest SAI

In this section, we describe two algorithms that are used to determine the correct amount of cable and structures that are necessary to connect each microgrid to the nearest SAI. When FEEDDIST is run in a fully optimizing mode, it computes the cost of distribution plant for all clusters using both approaches, and selects the approach giving the lower cost.[16]



[16] In order to generate approximately optimal results using less computing time, the user has the option of computing distribution costs using both approaches only for the lowest density grids.

Figure 3: Connection of Cells to the Closest SAI

The first algorithm is most appropriate in densely populated clusters, in which the proportion of populated microgrids to total microgrids is relatively large. Backbone cables run along every other cell boundary, and connect with the distribution plant within a cell at various points as illustrated in Figure 3. The location of the SAI divides the cluster into four quadrants. Beginning with the "southeast" quadrant, backbone cable is run along the boundary of the first and second rows of cells, running eastward until it reaches a point directly below the SAI. Cable from cells in row one is directed upward toward the cell boundary, while cable from cells in row two feeds downward. From the cell boundary point directly below the SAI, vertical cable then completes the connection to the SAI. By continuing along every other cell boundary, but extending cable only to populated cells, it is possible to connect every customer in the quadrant to the SAI. Similar connections apply to each of the remaining quadrants.

The second algorithm generally gives a more efficient distribution network for clusters with a lower population density, where the number of populated microgrids is smaller. In this case, the construction of an optimal distribution network within a cluster is closely related to the problem of constructing an optimal feeder network for the entire wire center, and we are able to use the same algorithm to provide a solution. In this approach, the algorithm described in the Feeder Plant Design section below (based on Prim, 1957) is used to connect each drop terminal placed within each microgrid (as shown in Figure 2) to a network consisting of all drop terminal locations and all SAIs.

HCPM always computes a distribution cost using the first algorithm. Since computations involving the Prim algorithm can be time consuming for large clusters, it is recommended that the second approach be used only for lower density clusters. Whenever both algorithms are used, distribution cost is determined by choosing the minimum cost obtained.

3.2.2Feeder Plant Design

The model first computes the cost of each possible configuration of primary and secondary SAIs within a cluster, and selects the least cost option. If a secondary SAI is used it is connected to the primary SAI by T1 lines using a minimum distance spanning tree network using an algorithm described in the following section. For each configuration of 1 or 2 SAIs, the model computes the total cost within the cluster of the distribution plant which connects each customer to the closest primary or secondary SAI; the T1 connections which link all primary and secondary SAIs; and the cost of all associated terminals. Then, based on the locations of the central office and all primary SAIs within a quadrant, it determines the feeder network which connects each of the primary SAIs to the central office.

The cost of copper-based T1 terminals and fiber-based DLC (digital loop carrier) terminals are determined as follows. The terminal sizes are a function of the line capacities required. The digital signal hierarchy, typically referred to as the DS-n where n is 1, 2 or 3 (and multiples of 3 referred to as OC-n), is used to size the terminal requirements. Each DS-1 or T1 may be used to support 1-24 lines. A DS-3 or T3 may be used to support 672 lines (28 DS-1's). Finally, an OC-3 (3 DS-3's) may be used to support 2016 lines. Five terminal sizes with line capacities of 2016, 1344, 672, 96 and 24 may be used. Fiber terminals require 4 fibers per terminal. T1 terminals, with line capacities of 24 or 96 require two copper pairs per DS1 line: one for transmitting signals from the CO to the primary SAI, and one for receiving signals from the primary SAI at the CO. A user adjustable 4 to 1 redundancy ratio is assumed in the model, which means that 10 copper pairs are required to serve each 96 line T1 terminal.

The minimum cost obtained defines the total distribution and internal feeder cost for the cluster, and the optimal number of primary and secondary SAIs. At this point the model has defined for every cluster the cost minimizing collection of primary and secondary SAIs, and has computed the total cost of connecting every customer to the closest *primary* SAI.

Feeder and Subfeeder Routes

In previous versions of the HCPM, and in other models of the local exchange (e.g. Gabel and Kennet, 1991) feeder plant was deployed in a "pine tree" network in which four main feeder routes emanate from the central office along East-West and North-South routes. Subfeeder routes perpendicular to the main feeder routes were then used to bring the feeder system closer to individual SAIs. This design proved to be highly efficient in terms of creating opportunities for the sharing of structure costs among feeder cables serving different SAIs. Lower structure costs made possible by increased sharing, however, came at the expense of longer feeder routes and correspondingly higher cable costs. In order to balance these two opposing tendencies, the HCPM examined a large number of possible feeder systems having different number of subfeeder routes, and chose the configuration giving the lowest cost.

The current version of HCPM uses a variant of an explicit optimization algorithm, discovered by Prim in 1957, to determine the trade-off between structures and cable costs.[17] This algorithm is based on some well known mathematical principles of network design based on techniques of discrete mathematics and graph theory. An abstract network consists of a single "supplier" node, a set of customer nodes, a cost function specifying the cost of connecting any two nodes, and a set of pairwise traffic demands between any two nodes. In the application of the Prim algorithm to the feeder network, the supplier node is the central office for a given wire center, and the customer nodes are the remote terminals, or SAIs, that define the interface points between the feeder and distribution portions of the network. The algorithm can also be applied to determine cable routes for distribution networks within a cluster as noted previously. In this case, the supplier node is an SAI within a cluster and the customer nodes represent individual subscriber locations that are to be connected to that SAI.

In both the feeder and distribution portions of the network, the objective of the telecommunications engineer is to minimize the cost of connecting each customer node to the supplier node. While in general this is an extremely difficult problem to solve, there are several special cases in which efficient algorithms exist which define a fully optimal network solution. One special case of interest is the construction of a "minimum distance spanning tree network" in which the sole objective is to minimize the aggregate length of communications links within the network. Such a network would be approximately optimal when traffic demands are sufficiently low that the actual cost of each link in the network is largely determined by the cost of structures (which depend only on distance).

A minimum distance network can be constructed using the Prim algorithm in the following way. Beginning with a network consisting only of the supplier, find the nearest customer node that is not yet attached to the network and attach it. The network then consists of the supplier and one customer. The algorithm proceeds step by step in attaching customer nodes to the network on the basis of minimum distance. At any point in the algorithm, it chooses from the set of all nodes that have not yet been attached, a node that is closest to some node in the existing network. Prim demonstrated that this simple algorithm necessarily leads to a minimum distance network.[18] In other words, when the algorithm is completed, there is no possible way to reconfigure the network so as to lower the aggregate distance of all links in the network.

As long as structure costs are significantly larger than cable costs, the original Prim algorithm provides a satisfactory solution. In the design of both feeder and distribution networks, however, a minimum distance spanning tree network is not generally optimal.[19] While it minimizes the total distance

[17] See Prim, R.C. (1957), "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal, 36*, 1389-1401 for a description of an efficient algorithm for computing minimum distance networks. A computed coded version of the Prim algorithm, and some extensions, is contained in Gower, J.C. and G.J.S. Ross (1969), "Minimum Spanning Trees and Single Linkage Cluster Analysis," *Applied Statistics, 18*, 54-64.

[18] In fact, one can start with any initial node and be assured of reaching a minimum distance network using the algorithm.

[19] The unmodified Prim algorithm is, however, used to connect multiple SAIs within a grid and for connecting drop terminal nodes to SAIs. In the mathematical literature on network design, networks which

of all links in the network, it does not minimize the distance between any particular node and the supplier. For example, if there is significant demand at a particular remote terminal for access lines to the central office, then the actual cost on the network between this node and the central office may need to be accounted for in the minimum distance optimization problem. There are no simple algorithms which can be applied to give a completely optimal solution to the general problem. (A "star" network, which minimizes the cost of connecting each node to the central office would not be optimal because it does not take advantage of potential sharing of structure costs in the network.) However, it is possible to modify the Prim algorithm to take account of the effects of traffic in the network on total cost, and generally to improve the performance of the algorithm computationally.

The HCPM makes two fundamental modifications to the Prim. First, the model creates a set of potential junction points which follow the location of the main East-West and North-South feeder routes in previous versions of the model. The algorithm permits, but does not require, these potential junction points to be used in creating the feeder network. Junction points create additional opportunities for the sharing of structure costs, and in some circumstances they can also reduce the distance between a terminal node and the central office, as the examples in Figure 4 demonstrate.[20]

The second modification of the Prim algorithm is in the rule which is used to attach new nodes to the network. Rather than minimizing the distance from an unattached node to the existing network, the algorithm minimizes the total cost of attaching an unattached node, and of constructing all of the lines that are required to carry traffic from that node back to the central office. A heuristic description of the algorithm is given below.

Step 1: Begin with a network consisting of the central office alone.

Step 2: From the set of unattached nodes, find the node for which the average cost per line, including cost of structures, cable and terminal electronics is lowest for connecting that node to the existing network.

Step k: At any step in the algorithm, choose from the set of unattached nodes the node for which the average cost is lowest for connecting that node to the existing network. This cost will depend on the particular node in the existing network that is selected for connection, and will include the structure cost of connecting to that node as well as the incremental cable cost of carrying traffic from the new node to the central office along the currently existing network. Repeat step k for all new nodes.

The algorithm terminates when all nodes have been attached. Unlike the Prim algorithm, it may be possible to lower total network cost by rearranging some of the links in the network after the algorithm terminates. However, the general optimization problem is computationally intractable, while the above algorithm is highly efficient. We have found that this modified Prim algorithm leads to lower feeder cost estimates than the unmodified Prim algorithm and the more traditional pine tree feeder route designs. Furthermore, we believe that the modified Prim algorithm provides a good approximation to the way in which real world engineers are likely to design the feeder network, since the network grows naturally from the central office, by adding new nodes on the basis of minimum attachment cost as new communities are established.

In the construction of the feeder network, the HCPM allows the user to determine whether to use airline distances between nodes or rectilinear distances. The model also applies a road factor to all distance computations in the feeder network. This road factor is intended to convert distances determined by the distance function into actual route distances, which must account for the existing road network and other terrain factors. In principle, the road factor should be determined empirically for each region of the country by comparing actual feeder route distance to model distance computation. Clearly a different factor should be applied to airline distance than to rectilinear distance computations. Some empirical evidence on the appropriate value for the road factor is given in Love et al. (1987).

---

allow for the creation of junction pointsare known as "Steiner networks." [See Sharkey (1995)]. A Steiner network must always have a cost at least as low as a minimum distance spanning tree network.

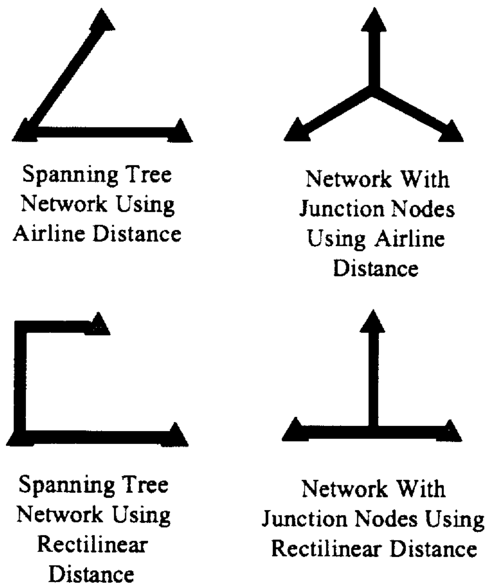


Figure 4: Minimum Distance Networks With and Without Junction Points

The figures below illustrate the operation of the Prim algorithm and the HCPM modifications to it. We randomly generated a set of 15 locations representing one wire center (source) and 14 customer nodes. Each customer node was assigned a demand of one unit. We examined a highly simplified cost function of the form *Cost per link* = $(P_S + T * P_C) * distance$, where P_S represents the price per foot of structures, P_C represents the price per foot of cable, and T represents the traffic carried on the link. In Figure 5a we set $P_S = 1$ and $P_C = 0$. Since the only cost is the distance related cost of structures, this example illustrates the outcome of the unmodified Prim algorithm. In Figure 5b we set $P_S = 0$ and $P_C = 1$. In this case, the algorithm seeks to minimize the distance of every node from the central office, resulting in a "star" network. In Figure 5c we set $P_S = 1$ and $P_C = 1$. This example illustrates a balanced network that would be constructed if cable costs and structure costs are both significant cost drivers. Figure 5d illustrates a balanced network assuming rectilinear distances rather than airline distance. Figures 5e and 5f represent the effect of creating possible junction points along the North-South and East-West axes emanating from the central office.

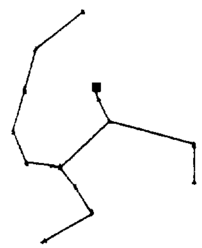


Figure 5a: Minimum Structure Distance Network

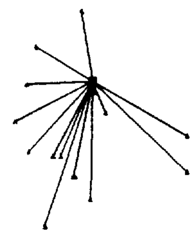


Figure 5b: Star Network

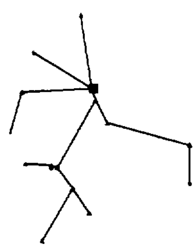


Figure 5c: Balanced Network

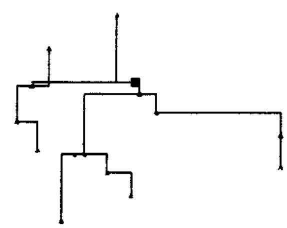


Figure 5d:: Balanced Network with Rectilinear Distance

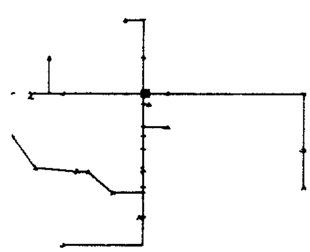


Figure 5e: Balanced Network with Junction Nodes and Airline Distance

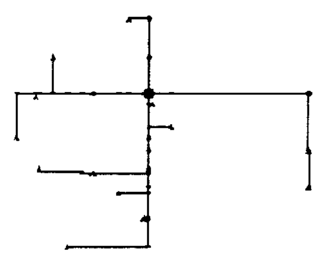


Figure 5f: Balanced Network with Junction Nodes and Rectilinear Distance

Figure 6 illustrates the feeder network constructed by FEEDDIST for a wire center in Montana. Solid circles represent SAIs; open circles represent junction nodes and diamonds represent the center points of all populated microgrid cells.

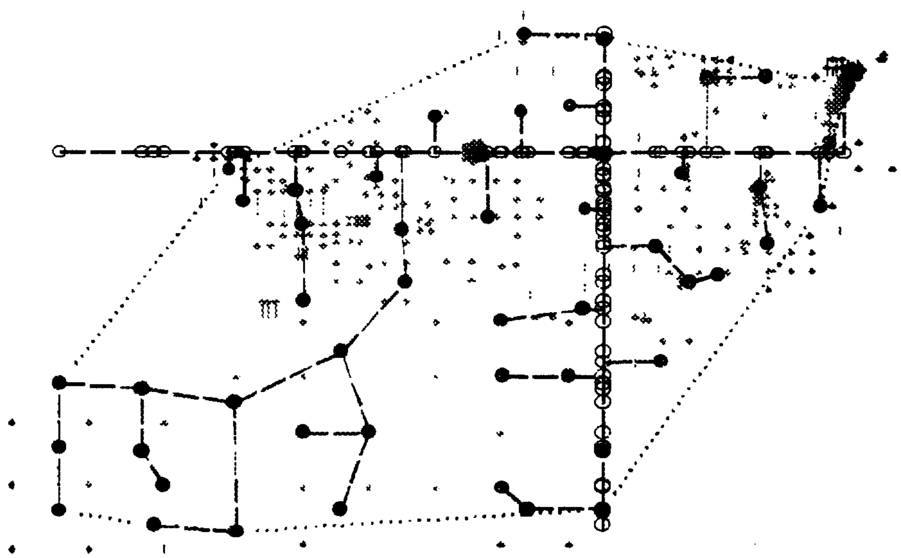


Figure 6: Feeder Network for the Wire Center ABSRMTXC

4. Comparison of HCPM and the Industry Sponsored Models

In this section we compare the HCPM customer location and loop design modules with comparable modules from each of the alternative industry supported proxy models. As noted in sections 2 and 3, HCPM differs most significantly from the other models in two respects. The customer location algorithm in HCPM is capable, in principle, of building loop plant to exact customer locations. As a practical matter, with existing computer resources, the model aggregates customers into microgrid cells which measure 360 feet on a side. Assuming that an accurate set of geocoded customer location data is available, the model will introduce additional error only to the extent that customers are relocated within these microgrid cells. In contrast, both the HAI and BCPM models assume that customers are uniformly distributed in significantly larger rectangular regions, with the result that true customer dispersion can be significantly distorted in the modeling process. We will demonstrate in this section that these models do in fact introduce a systematic bias in the way that customers are located.

The second significant difference between HCPM and each of the alternative models is the degree of cost minimization that is achieved internally within the model. The HCPM does not rely on approximations of forward-looking network design principles which are built into *ad hoc* algorithms of the model. Rather, the model uses algorithms that consistently seek to minimize network investment based on input prices supplied to the model. When input prices change, the fundamental design of the network is capable of changing in response. Since the relative prices of critical inputs, particularly the price of circuit equipment, can be expected to change over time, we believe that the internal optimization routines that are part of the model will make HCPM a far more flexible tool which can be used to re-evaluate forward-looking costs as input prices change. We will document some of these results in this section.

4.1 Customer Dispersion

As noted above and in section 3.1, the HCPM attempts to build plant as closely as possible to individual customer locations. Customer location data is provided to the model via flat files containing the exact latitude and longitude for each residential and business location in a wirecenter. After the clustering module groups these locations into clusters, which will represent distribution serving areas, a grid overlay process us used to locate customers within microgrids. The loop design module then builds plant directly to microgrid cells using either a simple algorithm that follows the lattice lines of the grid structure, or using the spanning tree algorithm described in section 3.2. A stylized representation of this process is illustrated in Figure 7.

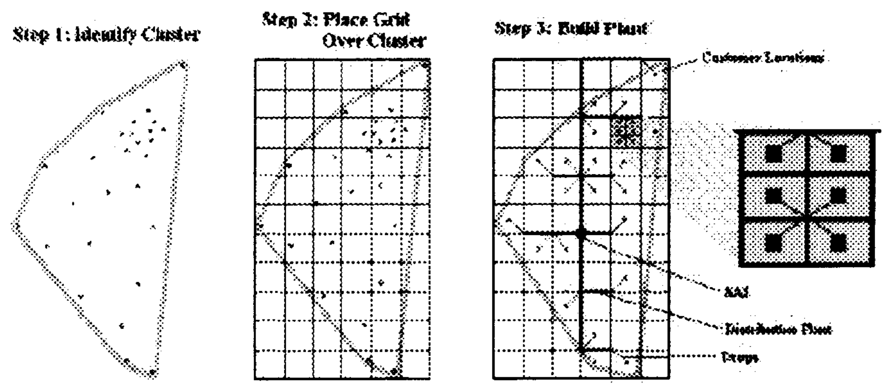
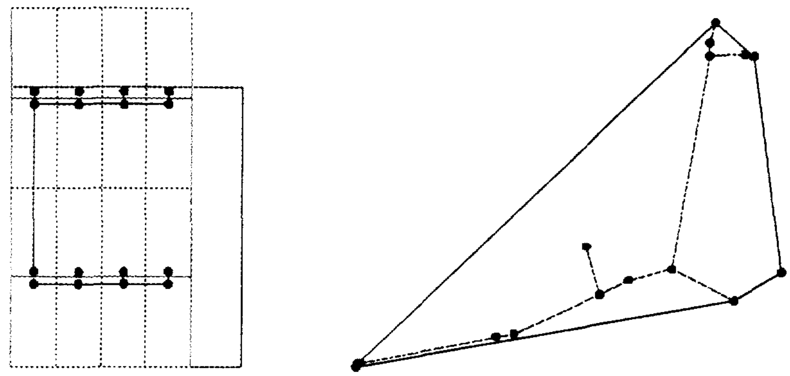


Figure 7: HCPM Grids Overlayed on Clusters

The HAI model converts outputs of its clusting module into a form that can be used by its loop design module in a different manner. First, a bounding rectangle is constructed around each cluster and an "aspect ratio" is computed which represents the ratio of adjacent sides of this rectangle. Next a rectangle with the same aspect ratio as the bounding rectangle, and an area equal to the area of the convex hull of the cluster is constructed. Customers are located more or less uniformly within the second rectangle, with the constraint that "lots" are created for each location in which the length of a lot is exactly twice the width. In Figure 8, the final locations of customers based on the hypothetical cluster containing 16 customer locations shown at the right are illustrated in the left portion of the figure. These manipulations are carried out in the pre-processing phase of the model in order that the run-time version, which computes the cost of loop plant in an Excel spreadsheet can perform the necessary computations in a reasonable period of time. A result is that customers are moved substantial distances from their original locations, and the set of locations that the model actually builds plant to are configured in a more uniform manner than the underlying data dictate. We will, in fact, demonstrate that this transformation of customer locations leads to a systematic bias in the cost of providing service to customers in low density serving areas.



Locations Used to Construct Loop Plant | **Customer Locations in Hypothetical Data Set**

Figure 8: HAI Customer Location Within Clusters

In order test for possible bias in the HAI customer location algorithms, we randomly generated a series of points, N, which all lie within a 18 kft by 18 kft box. Each set of points represents a hypothetical cluster of customer locations that could be served by a single SAI. We next calculated two different measures of dispersion for these points by constructing a star network connecting each point in the cluster directly to the centroid, and by constructing a minimum spanning tree network connecting all of the points. Next we applied the HAI algorithm to the initial set of points and calculated the length of the star network and of the minimum spanning tree network for the HAI points. We performed 2440 trials of this experiment and grouped the trials according to the number of points, N, in the cluster. Since the size of the cluster is held constant, the greater the number of customer locations, the greater the density of the cluster. We computed an error rate for each trial, where a negative error indicates that the HAI algorithms underestimate the dispersion of the original points, where dispersion is measured in terms of both the star network and the minimum spanning tree network. For example, in trials with N equal to 25, we found that the HAI algorithm underestimates the length of the star network by an average of 15.4 percent. It underestimates the length of the minimum spanning tree network by an average of 41.5 percent. These underestimates correspond to error rates of -0.154 and -0.415, respectively. In order to represent these results for all values of N, we computed the kernel smooths of the data that were generated. Kernel smoothing is a non-parametric regression technique that allows one to plot an underlying trend line given "noisy" data.[21] Using a uniform kernel to estimate the HAI algorithm's error rate as a function of N (equivalent to using an unweighted five period moving average) we obtained the results illustrated in Figure 9. These results indicate that the HAI approach systematically underestimates the dispersion of customer locations in low density areas. A similar analysis over hypothetical clusters with larger number of points, N, reveals that the HAI approach systematically overestimates customer dispersion in high density areas.

[21] For more information about non-parametric regression and smoothing see: Manski, C.F., March 1991, "Regression," *Journal of Economic Literature* XXIX: 34-50, and Härdle, W., 1989, Applied Nonparametric Regression, Cambridge: Cambridge University Press.

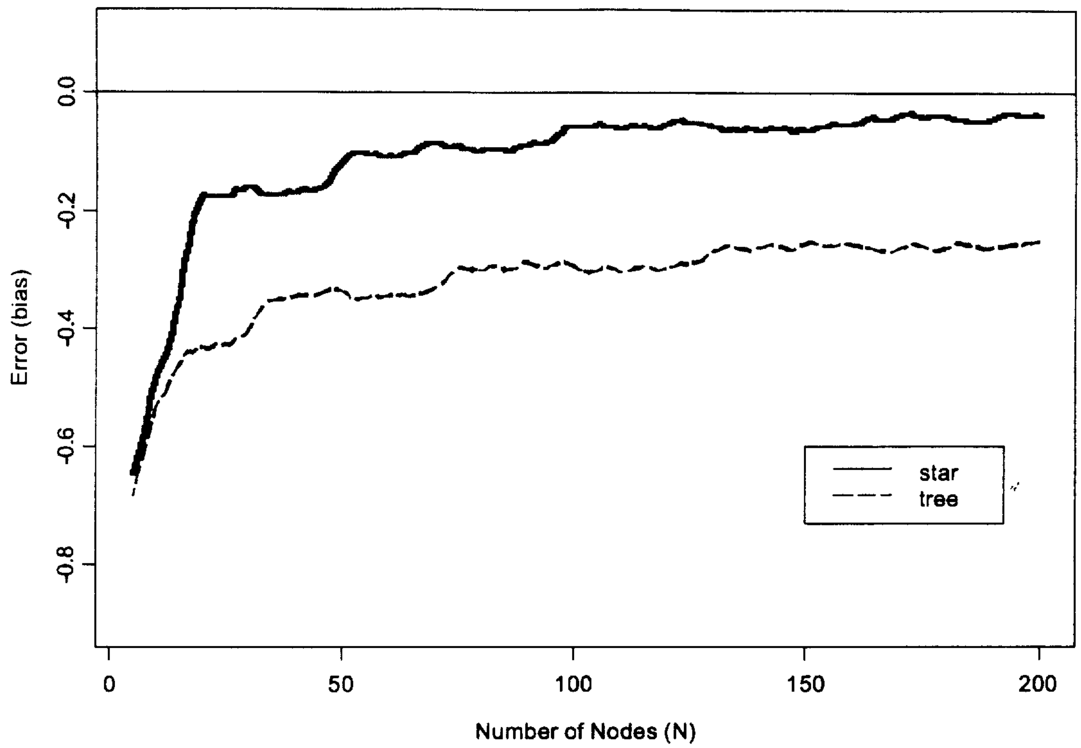


Figure 9: Kernel Smooths of Error in HAI Customer Location Approach

The BCPM model also introduces distortion in the customer location process. This model does not rely on a clustering algorithm of underlying data. Instead it uses Census population data along with Census Bureau data on the road network to determine the size and location of rectangular serving areas. The model seeks to create grids which would have approximately the same number of target customer lines as the HCPM or HAI cluster algorithms. Each grid is then subdivided into four quadrants. The origin of the coordinate axis which determins the quadrants is given by the centroid of the road network within the grid. Within each quadrant, a rectangle is constructed whose area is equal to the length of all roads within the quadrant multiplied by 1000 feet. The logic of the latter computation is that population can be expected to live within 500 feet of roads. Within each of the final rectangles in any quadrant, customers are located in a uniform fashion much as in the HAI module. The BCPM approach to customer location is illustrated in Figure 10.

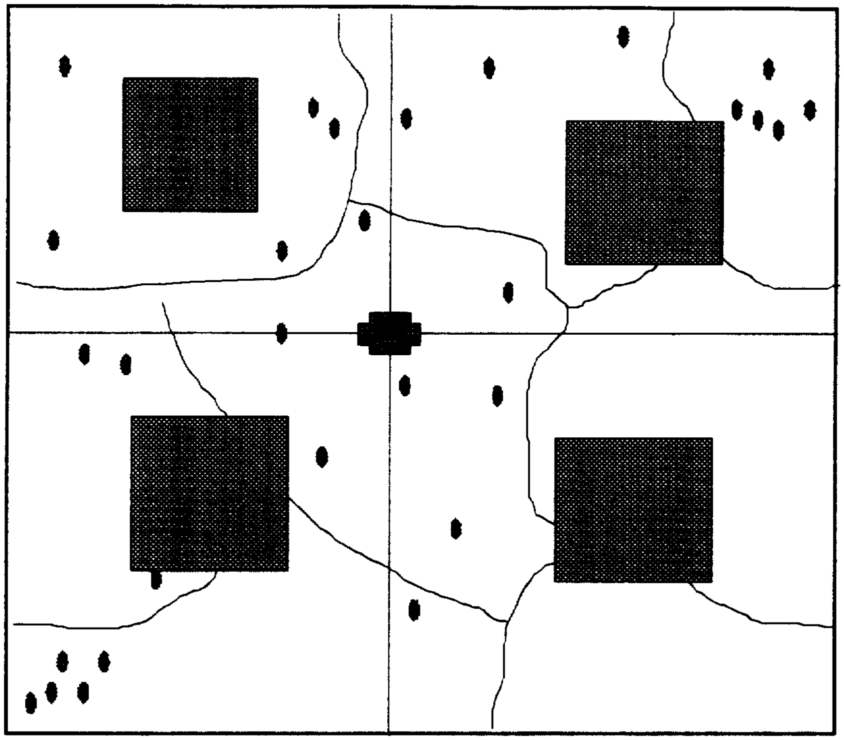


Figure 10: The BCPM Approach to Customer Location

Since we do not have access to the algorithms or the road data used by the BCPM model, we cannot perform a quantitative test of locational bias that is comparable to the reported test of the HAI model. We nevertheless believe that similar results would be obtained in such a test.

4.2 Optimization

The second significant difference between HCPM and both the BCPM and HAI models is the sophistication of the algorithms used to approximate a cost minimizing solution to the network design problem. The HCPM can be used to estimate the costs of networks designed for different jurisdictions or to different quality standards. For example, under one set of input specifications the HCPM will specify a maximum distance of 18 kilofeet from any customer location to the corresponding cluster centroid, and a maximum analog copper distance of 12 kilofeet if 26 gauge copper is used. The former specification seeks to determine the outer boundaries for a serving area based on currently available technologies, while the latter specification conforms to the current Bellcore standard for analog copper loops capable of supporting voice band services (POTS).[22]

The model can be easily adapted, through user input decisions, to accommodate alternative design specifications. For example, the suggested cluster radius could be increased beyond 18 kilofeet if technology advances suggest the feasibility of doing so, or it could be reduced to 12 kilofeet or less in order to accommodate alternative network design principles. Similarly, the user can independently adjust the 12-kilofoot maximum copper distance in order to accommodate technological advances or differing opinions on the appropriate quality standard.

[22] See "BOC Notes on the LEC Networks", Bellcore Special Report SR-TSV-002275, April, 1994. Longer copper loop lengths can be accommodated using a mix of 24 and 26 gauge cable.

For each serving area, and each feasible technology (analog, T1, or fiber) the model selects the number of T1 terminals and/or fiber terminals based on integer optimization. In addition, the model optimizes over technology type during the construction of the feeder and sub-feeder system, since the distance of each SAI from the central office is a function of the feeder system under consideration. The model selects technologies (e.g. fiber vs. copper, aerial vs. buried) on the basis of annual cost factors which account for both operating expenses and capital expenses over the expected life of the technology.

In this section, we illustrate the potential impact of optimization within the model by conducting a sensitivity analysis of the results for a medium size company as the price of circuit equipment changes. Since the FCC synthesis model, which incorporates the HCPM loop design modules, has been designed to make use of the HAI expense module, a direct comparison between the synthesis model and the HAI Model, version 5.0a is possible. To conduct our test, we use the HAI recommended default set of input values in the HAI model, and we calibrate the HCPM specific input values for prices of cable and structure to the HAI defaults as closely as possible. Both models were run for the Roseville Telephone Company of California to establish a base case. To determine the effects of a substantial reduction in the price of electronics, we reduced the prices of terminal electronics equipment in each model by a factor of 10.[23] A reduction in electronics prices should be expected to lead to a substitution of fiber feeder plant for copper plant in a well designed network which adheres to cost minimization principles. However, to the extent that a model relies on engineering rules of thumb, the possibilities for such substitution might be limited. Of course, copper feeder plant can only be utilized for clusters that are located relatively close to the serving wirecenter – a distance of 18,000 feet by default in both the HCPM and HAI models.

Since the price of copper feeder cable is identical in both simulations for each model, one can precisely determine the degree of substitution away from copper feeder plant by comparing investment dollars for copper feeder before and after the price reduction. Since the price of electronics is different by a factor of 10 after the price reduction, one can determine the percentage increase in the number of fiber terminals by multiplying the total investment in electronics after the price reduction by 10 and subtracting the initial investment. We obtained the following results:

	HCPM default	HCPM low DLC	HAI default	HAI low DLC
Copper feeder investment	$9,210,630	$78,008	$2,397,772	$1,472,758
Electronics investment	$7,210,615	$2,476,949	$12,442,630	$7,506,750
DLC lines	38,345	130,356	76,892	87,980
Total loop investment	$53,369,550	$39,742,766	$60,864,341	$54,747,492

In the HCPM model a tenfold reduction in the price of digital electronics leads to the substitution of virtually all (99%) of the copper feeder plant deployed in the default case, and a greater than threefold increase in the number of lines served by electronics. In contrast, in the HAI model, a similar price reduction leads to only a 38% reduction in copper feeder plant and a 14% increase in lines served by electronics. Overall, total loop investment falls by more than 25% in the HCPM model as a result of the DLC price reduction. In the HAI model, total loop investment falls by 10%.

5. Conclusion

In this paper we have described the structure and operation of an innovative computer based model of the local exchange telephone network known as the HCPM. The HCPM can be used in a variety of regulatory arenas, since it provides a regulatory agency with an independent source of information about the forward-looking costs of providing local telephone service. The model can also be used more broadly by governmental agencies in planning for infrastructure development to expand telephone service to currently unserved areas, or as in the case of the U.S. universal service program, in designing appropriate

[23] In HCPM all DLC costs in the input file "fdcost.txt" were multiplied by 0.1. In HAI, comparable costs in the window for DLC costs (located under feeder costs) were multiplied by 0.1.

subsidies for current providers in order to allow a re-balancing of rates between high cost and low cost regions of a country.

The HCPM represents an advance over previous models in its ability to build plant to precise customer locations. Given an accurate source of geocoded customer location data, the model can estimate the minimum cost of providing local loop plant that is sufficient to provide any desired quality of service. The model is also extremely flexible, in that good results can be obtained even with more highly aggregated Census level data.

While the model will require constant updating as technology changes (e.g. by incorporating a model of wireless local loop provision) the built in optimization routines should make the cost estimates produced by the model extremely robust to changes in input prices. Clearly the outputs of any cost model such as HCPM depend critically on the input values that it uses. In Appendix A the input set approved by the FCC in its October 1999 universal service order are described. Appendix B describes the basic operation of the model as well as the preparation of customer location data sets.

Appendix A
Computing the Cost of Distribution and Feeder Plant

In this section we describe how the HCPM computes the cost of each element in the distribution and feeder plant based on the customer location and loop design algorithms described in section 3. The input values reported in this section represent inputs adopted by the Federal Communications Commission in its October 1999 universal service order. We do not, however, provide a comprehensive listing of all inputs that are required to run the model.

A.1 Cable Sizing and Costing

First we determine the appropriate technology type for the cluster. The customer location module reports for each microgrid the number of households and the number of business lines demanded. To convert these inputs into a demand for lines, we apply a residential line multiplier, given in Table 16, to the household count. We also adjust the business line count to take account of special access lines and the fraction of business lines (both special access and switched) that are carried on DS1. The sample values for these user adjustable inputs are given in Table 16. The effect of the latter adjustment is to reduce the number of copper pairs required in the distribution plant (and feeder plant when analog copper feeder is used) by 0.9167 times the number of business lines (DS0 equivalents) carried on DS1.[24] However, when accounting for the cost of remote terminals appropriate account must also be taken of the difference in cost between DS0 and DS1 line cards.[25]

In order to determine the correct size of cables at every point in the distribution network, we first locate the corner of the cell that is closest to the SAI and which is also located on the horizontal cell boundary that is used for distribution backbone cable. We begin at the opposite corner of the cell at the intersection point of four lots. Following the lot boundary toward the horizontal backbone cable we begin accumulating lines and cable. If we accumulate enough lines for a new cable, we add it. This exercise is repeated until we reach either the midpoint of the cell boundary containing horizontal backbone cable or the corner of the cell closest to the SAI. At these points, cables from all cells served by the horizontal backbone are merged and the correct size computed to continue on the path back to the SAI. (See Figures 2 and 3.)

Every customer is connected to the closest SAI by analog copper cable. If the maximum distance is less than or equal to the copper gauge crossover (user adjustable with a recommended value of 12 kf), then 26-gauge cable is used. If the maximum distance is greater than this threshold, then a combination of 24 and 26 gauge cable is used.[26] Cable sizes and sample values for the cost of 26-gauge copper are given in Table 1. Similar tables give the cost of 24 guage distribution cable and for both 26 and 24 gauge feeder cable.[27]

---

[24] This computation follows from the fact that two copper pairs providing providing a T1 service can substitute for 24 analog copper pairs.

[25] This is not implemented in the model at this time.

[26] The model places SAIs to ensure that the distance between a customer and the nearest SAI is always less than or equal to the copper distance threshold, which is set by default at 18 kf. Since the model optimizes the number and location of SAIs within a grid, it is able to examine the trade-off between fiber or T1 versus analog copper, and to choose the least cost alternative.

[27] After each table number the corresponding name of the data input file that is required to run the loop design module, FEEDDIST, is given in parentheses.

Table 1 (26gdist.txt) 26 Gauge Copper Distribution Cable Cost (per foot)			
Size	UG	Buried	Aerial
4200	$30.07	$31.81	$28.48
3600	$26.37	$27.37	$24.63
3000	$22.67	$22.93	$20.78
2400	$18.97	$18.49	$16.94
2100	$17.12	$16.27	$15.01
1800	$15.27	$14.05	$13.09
1200	$11.60	$9.61	$9.23
900	$9.78	$7.39	$7.31
600	$7.98	$5.17	$5.38
400	$6.82	$3.68	$4.08
300	$6.27	$2.94	$3.44
200	$5.75	$2.20	$2.78
100	$5.32	$1.45	$2.12
50	$5.18	$1.07	$1.79
25	$5.15	$0.88	$1.61
18	$5.15	$0.82	$1.57
12	$5.15	$0.78	$1.52
6	$5.16	$0.73	$1.48
1	$5.16	$0.69	$1.45

In order to connect a customer to the distribution network, a network interface device (NID), drop wire, and a drop terminal are required. The sample values for the cost of the NID and drop wire are reported in Table 16. The cost of drop terminals is given in the following table. If more than 25 lines need to be placed at a single location, the model assumes that it is a business location, and the costs in each column of the table represent the cost of an indoor SAI.

Table 2 (drop.txt) Drop Terminal Cost			
Size	Buried	Aerial	UG
1	$133.46	$70.44	$133.46
6	$157.05	$95.98	$157.05
12	$440.87	$131.81	$440.87
25	$451.00	$216.00	$451.00
50	$220.00	$220.00	$220.00
100	$333.00	$333.00	$333.00
200	$665.00	$665.00	$665.00
400	$1,331.00	$1,331.00	$1,331.00
600	$1,996.00	$1,996.00	$1,996.00
900	$2,770.00	$2,770.00	$2,770.00
1200	$3,993.00	$3,993.00	$3,993.00
1800	$5,539.00	$5,539.00	$5,539.00
2400	$7,536.00	$7,536.00	$7,536.00
3600	$11,079.00	$11,079.00	$11,079.00
5400	$16,618.00	$16,618.00	$16,618.00
7200	$21,708.00	$21,708.00	$21,708.00

Cable sizing algorithms similar to those used for distribution plant are used to determine the appropriate size of cable along the feeder and subfeeder network. In the case of fiber, an analysis of the cost of performing a fiber splice is conducted, and the model determines to either splice the fiber or to maintain separate cables at each junction point. Copper and T1 cables are currently assumed to be spliced at junction locations. If analog copper cable or T1 on copper are deployed in the feeder plant, it is assumed that 26-gauge copper is used, and cable costs are reported in Table 1. When fiber cable is deployed in the feeder plant, its costs are given in the following:

Table 3 (fibrcabl.txt) Fiber Feeder Cost (per foot			
Size	UG	Buried	Aerial
288	$9.43	$8.89	$8.17
144	$6.14	$4.76	$4.69
96	$5.04	$3.39	$3.53
72	$4.49	$2.70	$2.95
60	$4.22	$2.35	$2.66
48	$3.94	$2.01	$2.37
36	$3.67	$1.67	$2.08
24	$3.40	$1.32	$1.79
18	$3.26	$1.15	$1.65
12	$3.12	$0.98	$1.50
1	$2.87	$0.66	$1.24

Cable distances in the feeder plant are adjusted by a series of factors. The model adjusts feeder distances by factors which depend on the maximum and minimum slope within a cluster. (All factors and triggers are listed in Table 16.) Specifically, if the customer location module reports that the maximum slope in a cluster is greater than the maximum slope trigger and the minimum slope is greater than the minimum slope trigger, then the combined slope factor is applied to feeder distance calculations. If only one of the slope triggers is exceeded, the corresponding slope factor is applied. The model also allows the user to specify a road factor, which converts model distance computations to an empirical estimate of road distance.[28]

Cable sizes at each point in the distribution and feeder network are adjusted by the following set of fill factors.

Table 4 (fillfact.txt) Fill Factors		
Density	Feeder	Distr
0	70.0%	50.0%
5	77.5%	55.0%
100	80.0%	55.0%
200	82.5%	60.0%
650	82.5%	70.0%

[28] Empirical values for the road multiplier are reported in "Mathematical Models of Travel Distances," Chapter 10, *Facilities Location: Models and Methods*, R.F. Love, J.G. Morris, and G.O. Wesolowsky. Amsterdam: North Holland (Elsevier), 1988. For the Euclidean distance metric, distance multiples of 1.16-1.35 are reported. For rectilinear distance, values range from 0.95 to 1.05.

850	82.5%	75.0%
2550	82.5%	75.0%
5000	82.5%	75.0%
10000	82.5%	75.0%

A.2 Terminal Cost

Fiber terminal costs are computed on the basis of input values, based on the cost of installing either high density DLC units with line capacities of 2016 or 672, or a low density DLC units with a line capacity of 96 or 24. These costs, which are reported in Table 16, include the costs of site preparation and power, the cost of an optical patch panel, and the initial cost for housing and multiplexing at the remote terminal.

If T1 on copper is used in the feeder network, we assume that terminal capacities of either 24 or 96 lines are possible. A 96-line terminal requires 5 copper pairs, assuming that one redundant pair for maintenance is provided for every four active pairs. The fixed and variable costs of T1 terminals are reported in Table 15.

In addition to the cost of terminal electronics there are additional costs for the physical interface between feeder and distribution plant. The costs of these units is given in the following table.

Table 5 (fdi.txt) Feeder Distribution Interface		
Lines	Outdoor	Indoor
1	$150.78	$150.78
50	$562.00	$220.00
100	$787.00	$333.00
200	$1,349.00	$665.00
400	$2,248.00	$1,331.00
600	$3,147.00	$1,996.00
900	$4,271.00	$2,770.00
1200	$5,395.00	$3,993.00
1800	$7,644.00	$5,539.00
2400	$9,667.00	$7,536.00
3600	$13,489.00	$11,079.00
5400	$18,434.00	$16,618.00
7200	$22,481.00	$21,708.00
9000	$30,125.00	$27,247.00
10800	$35,970.00	$32,787.00
12600	$40,915.00	$38,326.00
14400	$44,962.00	$43,416.00
16200	$52,606.00	$48,955.00
18000	$58,451.00	$54,495.00
19800	$63,396.00	$60,034.00
21600	$67,443.00	$65,124.00
23400	$75,087.00	$70,663.00
25200	$80,932.00	$76,203.00
27000	$85,877.00	$81,742.00
28800	$89,924.00	$86,832.00

A.3 Structure Costs

Structure costs are determined by the distance computations derived in section 3, and the following tables of user adjustable input values.[29]

Table 6 (normal.txt) Normal Structure Placement Cost (per foot)						
	Underground		Buried		Aerial	
Density	Feeder	Distr.	Feeder	Distr.	Feeder	Distr.
0	$1.86	$1.86	$0.77	$0.77	$1.51	$1.51
5	$1.86	$1.86	$1.54	$1.54	$1.51	$1.51
100	$7.63	$7.59	$3.24	$3.14	$1.98	$1.98
200	$8.16	$8.38	$4.26	$4.45	$1.98	$1.98
650	$8.90	$9.25	$5.20	$5.52	$2.27	$2.27
850	$10.23	$10.53	$5.51	$5.82	$2.27	$2.27
2550	$14.15	$14.23	$7.34	$7.42	$2.64	$2.64
5000	$27.79	$27.78	$9.02	$9.00	$2.72	$2.72
10000	$42.59	$42.57	$11.93	$11.91	$2.72	$2.72

Table 7 (softrock.txt) Soft Rock Structure Placement Cost (per ft)						
	Underground		Buried		Aerial	
Density	Feeder	Distr.	Feeder	Distr.	Feeder	Distr.
0	$5.78	$5.78	$1.40	$1.40	$1.80	$1.80
5	$5.78	$5.78	$2.17	$2.17	$1.80	$1.80
100	$9.04	$9.10	$3.81	$3.93	$2.35	$2.35
200	$9.99	$10.25	$5.04	$5.32	$2.35	$2.35
650	$11.02	$11.31	$6.42	$6.70	$2.68	$2.68
850	$13.24	$13.52	$7.03	$7.29	$2.68	$2.68
2550	$18.66	$18.74	$9.18	$9.27	$3.13	$3.13
5000	$38.86	$38.85	$11.70	$11.68	$3.21	$3.21
10000	$61.20	$61.19	$16.15	$16.13	$3.21	$3.21

Table 8 (hardrock.txt) Hard Rock Structure Placement Cost (per ft)						
	Underground		Buried		Aerial	
Density	Feeder	Distr.	Feeder	Distr.	Feeder	Distr.
0	$9.69	$9.69	$2.04	$2.04	$2.09	$2.09
5	$9.69	$9.69	$2.80	$2.80	$2.09	$2.09
100	$16.93	$16.84	$4.89	$4.89	$2.71	$2.71
200	$17.57	$17.57	$6.45	$6.37	$2.71	$2.71

[29] The default values for placement costs reported in this section have been derived from BCPM input tables by setting all sharing percentages equal to 100%. For aerial cable, cost per foot is computed by dividing the BCPM cost per pole by the BCPM pole spacing default. Default pole spacing in the BCPM is 250 feet in the lowest four density zones and 150 feet in the three highest zones. Guys are spaced from 500 feet to 1500 feet, and the cost of guys is accounted for in computing the total cost per pole. Similar computations apply to the Hatfield model defaults reported in section 7.

650	$18.63	$18.69	$7.67	$7.73	$3.10	$3.10
850	$21 40	$21 45	$8 65	$8.69	$3.10	$3.10
2550	$27.62	$27.63	$11.87	$11.87	$3.61	$3.61
5000	$58 18	$58.19	$15.71	$15.72	$3.69	$3.69
10000	$92.02	$92.02	$22.46	$22.46	$3.69	$3.69

Underground structures require ductwork in both distribution and feeder plant, and manholes in feeder plant. One duct per copper cable is required. One duct for fiber cable, if any, and one duct for maintenance are also required. Duct costs have a sample value reported in Table 15. The installed cost for manholes containing up to 9 ducts is reported in Table 9. For each additional set of 9 or fewer ducts there is an incremental manhole cost which is reported in the last row of the table. These costs can be converted into a cost per foot using the spacing assumptions in Table 10.

Table 9 (mhcost.txt) Manhole Costs			
Duct Capacity	Normal	SoftRock	HardRock
2	$1,436.50	$1,511.50	$1.586.50
4	$4,472.47	$4,652.47	$4,832.47
9	$5,176.00	$5,336.00	$5,496.00
99	$3,070.00	$3,150.00	$3,230.00

Table 10 (mhspace.txt) Manhole Spacing	
Density	Spacing(ft)
0	725
5	725
100	725
200	725
650	575
850	575
2550	575
5000	400
10000	400

Placement costs for structures depend on input variables from CLUSTER including the depth to bedrock, rock hardness, surface soil texture and the water table depth. Soil textures are converted to 0 or 1 (Table 11). If the depth to bedrock is less than the normal copper or fiber placement depth (Table 16) *and* rock type is "hard" then Hard Rock Placement tables are used to determine structure costs. If the depth to bedrock is greater than or equal to normal placement depth *and* the soil texture type is equal to 0, then Normal Placement tables are used. Otherwise, Soft Rock Placement is used. If the water table depth is less than the critical water depth (Table 16) then a cost multiplier based on the water factor (Table 16) is applied to all structure costs.

Table 11 (soiltex.txt) Surface Texture Table (Excerpt)	
Soil Type	Impact
BYX-FSL	1
BYX-L	1
BYX-SIL	1

BYX-SL	1
C	0
CB	0
CBA	1
CBA-FSL	1

The relative proportions of aerial, buried and underground plant vary by density zone according to the following tables: A similar table for copper feeder plant is not shown. The final cost of structures is also determined by the fraction of these costs assigned to telephone companies, which is a user defined inputs as illustrated in Table 14.

Table 12 (distmix.txt) Distribution Plant Mix			
Density	UG	Buried	Aerial
0	0.00%	60.00%	40.00%
5	1.00%	62.00%	37.00%
100	2.00%	68.00%	30.00%
200	4.00%	66.00%	30.00%
650	8.00%	62.00%	30.00%
850	20.00%	50.00%	30.00%
2550	40.00%	30.00%	30.00%
5000	60.00%	10.00%	30.00%
10000	90.00%	0.00%	10.00%

Table 13 (ffdrmix.txt) Fiber Feeder Plant Mix			
Density	UG	Buried	Aerial
0	5.00%	50.00%	45.00%
5	5.00%	50.00%	45.00%
100	5.00%	50.00%	45.00%
200	20.00%	40.00%	40.00%
650	40.00%	30.00%	30.00%
850	60.00%	25.00%	15.00%
2550	75.00%	15.00%	10.00%
5000	90.00%	5.00%	5.00%
10000	95.00%	0.00%	5.00%

Table 14 (sharing.txt) Structure Costs Assigned to Telephone			
Density	UG	Buried	Aerial
0	100.00%	100.00%	50.00%
5	100.00%	100.00%	50.00%
100	85.00%	85.00%	50.00%
200	65.00%	65.00%	50.00%
650	65.00%	65.00%	50.00%
850	65.00%	65.00%	50.00%
2550	55.00%	55.00%	35.00%
5000	55.00%	55.00%	35.00%
10000	55.00%	55.00%	35.00%

A.4 Miscellaneous Input Tables

The following tables describe sample input values that have not been previously documented.

Table 15 (fdcost.txt) Miscellaneous Cost Inputs	
Value	**Variable**
$560.00	cost_per_drop_kf
$39.50	nid_cost
$720.00	duct_cost_per_kf
$152,617.43	a2016
$74.98	b2016
$107,224.92	a1344
$74.98	b1344
$97,443.38	a672
$74.98	b672
$23,848.20	a96
$87.30	b96
$19,881.39	a24
$87.30	b24
$23,848.20	ac96
$87.30	bc96
$19,881.39	ac24
$87.30	bc24
$11,000.00	site_prep_cost
$0	fiber_splice_cost

Table 16 (feeddist.prm) Miscellaneous Engineering Inputs	
0.5	max_drop_length
0.5	user_lambda
12	copper_gauge_xover
18	max_copper_distance
1.25	MaxCopperPenalty
18	copper_t1_xover
0	t1_fiber_xover
1.25	t1_redundancy_factor
24	copper_placement_depth
36	fiber_placement_depth
3	CriticalWaterDepth
1.3	WaterFactor
12	MinSlopeTrigger
1.10	MinSlopeFactor
30	MaxSlopeTrigger
1.05	MaxSlopeFactor
1.20	CombSlopeFactor
12.7500%	pct_ds1
91.75%	pct_lsa

24	ChannelsPerT1System
2	PairsPerT1System
4	FibersPerTerminal
2016	CapacityF2016
1344	CapacityF1344
672	CapacityF672
96	CapacityF96
24	CapacityF24
96	CapacityT96
24	CapacityT24
10	lines_per_bus
1.00	DistRoadFactor
1.00	FiberFillFactor
1	DistanceType
1.00	FeederRoadFactor
2	Max_SAIs

In determining which technologies to deploy throughout the model, the decision is based on annual charge factors which take account of operating expenses as well as capital expenses. These factors are multiplied by the computed investment in each of the feasible technologies, and the decision is made to deploy the technology that will minimize costs over the expected life of the technology. The appropriate value for each factor depends on both expected maintenance expense and depreciation, and must currently be set by the user based on an external expense module or direct computations. In Table 17 below, the annual charge factors were computed on the basis of cost of capital and depreciation rates used in both the HAI and BCPM models, along with expense ratios derived from ARMIS data for all reporting local exchange companies. Since only the relative values of the annual charge factors are relevant, one can instruct the model to consider only first investment costs by setting all of the annual charge factors equal.

Table 17 (ancfg.txt) Annual Charge Factors	
Annual Factors	**Variable**
0.166762	ac_ugd_cop
0.192854	ac_bur_cop
0.224354	ac_aer_cop
0.150832	ac_ugd_fib
0.146548	ac_bur_fib
0.150824	ac_aer_fib
0.133737	ac_ugd_struc
0.169701	ac_bur_struc
0.182915	ac_aer_struc
0.133737	ac_manhole
0.191178	ac_t1_term
0.191178	ac_fib_term
0.189371	ac_fdi
0.142135	ac_fib_splice

Appendix B
HCPM Data Input Requirements

B.1 Parameter Inputs for CLUSTER

CLUSTER accepts user specified inputs from a file "cluster.txt" which can be used to modify inputs when the program is run in batch mode. The user can specify the particular clustering algorithm to use and the method and extent of optimization to apply to the clustering procedure. The user can also select input values that control the following additional parameters: Raster Size (in feet) represents the size of the cell that customer locations are assigned to prior to clustering; Distance Limit (in feet) specifies the maximum distance permitted from any customer location to the centroid of a cluster that contains that customer; Line Limit specifies the maximum number of customers permitted in a cluster; Line Fill specifies the percentage of the Line Limit that is filled during the initial phase of the Divisive algorithm.[30]; and Block divide factor controls the way in which Census blocks are divided when block level input data is used. The algorithm specifies that Census blocks are to be subdivided into units with areas less than or equal to $\left(\frac{\text{Distance Limit}}{\text{Block divide factor}}\right)^2$. Using the suggested inputs, Census blocks are therefore divided into squares no larger than 150 feet on a side, which is the default size of a raster cell.

Since the process of clustering can potentially take a long time when the number of customer locations is large, the cluster program attempts to keep the time devoted to clustering within manageable limits in various ways, which can be overridden by user input choices. The key user parameter which controls processing speed is the "Max pop cells" parameter, which is set by default equal to 1000. During the process of rasterization, the program assigns each of the individual customer locations specified in an input file to a raster cell. The target size for a raster cell is determined by the "Raster Size" variable. When the rasterization process is completed, a count of populated raster cells is done. If this number exceeds the specified maximum number of populated cells, the raster size is incremented by one unit.[31] This process continues until the number of populated cells is less than the user specified limit. When the divisive clustering algorithm is chosen, the maximum number of populated cells is set equal to "Max pop cells" multiplied by 3, since this results in roughly comparable time performance for the three alternative clustering algorithms.

B.2 Data Inputs for CLUSTER: <FILENAME>.IN

The currently recommended format for a cluster input file, <FILENAME>.IN, is a comma delimited ascii file. Each input file represents a single wirecenter with a single switch. Thus, the filename is typically the wirecenter code. The first line of the file should contain either the word "GEO" or the word "HOUSEHOLD" to identify the data aggregation level. (In both cases, however, the data points in <FILENAME>.IN represent individual customer locations. Input files with a "BLOCK" designation are assumed to contain contain Census block level data. Whenever block level data is detected, the cluster algorithm creates a new input file with surrogate locations and then processes this data as if it were true geocoded location dat.

In each input file, the second and fourth lines are header lines. The third line contains the wirecenter's CLLI code, the latitude and longitude of its switch, the latitude and longitude of its central

[30] As explained in section 3.1 this factor seeks to determine a good approximation to the cost minimizing number of clusters in more densely populated regions. The line fill factor has no effect on any of the clustering algorithms in sparcely populated regions. When the divisive algorithm is used, a line fill factor less than is recommended. Since both the agglomerative and nearest neighbor algorithms produce a larger number of clusters than the divisive, it is recommended that the line fill factor be set to 100% when these algorithms are used.

[31] If the number of populated cells exceeds "Max pop cells" when the raster size is equal to 500 feet, then a new rasterization is attempted with a raster size equal to 1000.

point, and the name of the company that provides it service. Starting on the fifth line, there is a record for each block or household. That record contains the following data: the Census Block number for that location, the Longitude and Latitude of the record's central point (if a Census block) or geocode location (if a point location), the number of residential lines at the location, the number of business lines, six fields containing terrain data; and the area associated with the location. The CB number consists of a sequence of digits which identify the State FIPS, the County FIPS, the Tract No., and the Block No. The Longitude and Latitude report the angular distance in degrees from the Greenwich meridan and from the equater respectively. The six fields containing terrain data can be empty, in which case these data are retrieved from a database as described below. When point data is used, the area field is set equal to 0. When Census Block data is used, area is reported in thousandth's of a square kilometer or equivalently in square meters divided by 1000.

An example of the first few lines of a valid input file for a wire center in Maryland follows:

```
HOUSEHOLD
Wc_code, SwX, SwY, CenX, CenY, Company
BRWKMDBR,-77.632272,39.321787,-77.604468,39.338588,BELL ATLANTIC - MARYLAND INC - MD
CBNum,Lon,Lat,,,,,,,,Area
24021752400101,-77.646338186,39.390221784,1,0,,,,,,0
24021752400101,-77.638810428,39.401627187,0,1,,,,,,0
```

The cluster module also makes use of a database in Microsoft Access 97 format called "Hcpm.mdb." This database contains two tables. The table "LineCount" contains a list of wirecenters identified by their Common Language Location Identifier (CLLI) code, followed by the number of residential locations (households), the number of business lines, the number of residential lines, the number of special access lines, the number of public telephone lines, and the number of single line business lines. The "Terrain" table contains a list of Census Block Groups (CBGs) followed by terrain data for each group consisting of bedrock depth, rock hardness, soil type, water table depth, minimum slope, and maximum slope.

B.3 FEEDDIST Input File Requirements

The parameter input files required for FEEDDIST are described in detail in section 5. In addition, CLUSINTF produces three binary output files which are used by FEEDDIST. When <FILENAME>.IN is used as an input to CLUSTER and CLUSTINTF, the files generated are named <FILENAME>.BIN, <FILENAME>.COO and <FILENAME>.DEN. The .BIN file contains, in Pascal record format, data required by FEEDDIST for each cluster. The .COO file contains the reference latitude and longitude used in distance calculations. The .DEN file contains the feeder route line density as calculated by CLUSINTF.

When FEEDDIST is called without an argument it prompts the user for the name of a wire center, and expects to find the input files associated with that name. When FEEDDIST is called with the argument <FILENAME> it automatically looks for the associated input files and begins processing. The first time that FEEDDIST is run it also creates a file FEEDDIST.CSV, which appends a row to a CSV file which describes a complete set of outputs for a particular wire center. This functionality makes it possible for FEEDDIST to run in batch mode for an entire study area. FEEDDIST can be run in batch mode by "batch" in place of <FILENAME> as an argument, in which case FEEDDIST will look for an ascii file named <BATCH.LST> consisting of a list of wire center names. Naturally, all of the input files associated with the file <BATCH.LST> must be present. When all of the wire centers in the batch have been processed, the user can then open the file FEEDDIST.CSV using any spreadsheet program and examine the results in detail.[32]

Three optional command line arguments to FEEDDIST can be specified. These may be entered in any order.

[32] The file FEEDDIST.CSV must not be open when FEEDDIST is run, or the program will terminate and an error message will be generated.

-c or -C:	Tells the program to close the window which appears on the user's screen after the program completes execution.
-v or -V:	Tells the program to generate additional outputs both on screen and in the following output files: <FILENAME>.OUT, DISTGRID.CSV, FEEDGRID.CSV, FEEDBYWC.CSV, and EXPENSE.CSV.
-pnnnn or -Pnnnn:	Tells FEEDDIST to utilize the Prim algorithm in all distribution areas where density is less than nnn. If nnn is not specified, the Prim algorithm will be used for all densities. If "-p" is not present on the command line, no Prim optimizing will be performed in the distribution plant.

The cluster interface program, CLUSINTF, also may be run with command line arguments. As in FEEDDIST, the user may specify either the name of a wire center or BATCH in the command line (or be prompted by the program after it launches). CLUSINTF flags are as follows:

-c or -C:	Closes window on program completion.
-snnnn or -Snnnn:	Sets raster size to nnnn kilofeet.
-w or -W:	Calculates population-weighted average density for clusters.

The -s flag should be used if a specific raster size is desired. Otherwise, the program will automatically calculate the raster size by dividing the maximum of (north-south distance, east-west distance)
for the cluster by 50. The program also will recalculate the user-set raster size if the implied number of raster cells in either direction exceeds 50; in this event, the raster size is calculated as above.

The -w flag should be used when the user anticipates that using the cluster density by itself leads to a misleading density estimate, such as would be the case when a high-rise residential development is located in the middle of a large farming region. If -w is not specified, density is calculated by dividing the number of lines in the cluster by the area of the convex hull of the cluster. If -w is specified, a population-weighted average of the default density with the overall wire center density is substituted.

B.4 Operation of the HCPM Interface

Version 2.6 of HCPM is distributed with a Visual Basic user interface. This interface automates the process of running all HCPM modules and allows the user to edit the user inputs through an Excel spreadsheet format. The interface also allows the outputs of FEEDDIST to be incorporated into the HAI model, version 5.0a for further processing by that model's switching, transport and expense modules. This interface and supporting documentation are available on the internet at

http://www.fcc.gov/ccb/apd/hcpm

The setup and operation of this interface is described separately in the document "The HCPM/HAI Interface for a Cost Proxy Model Synthesis: A User Manual" which is also available at the above site. The interface is supplied with all components from both the HCPM and HAI models that are required to run the synthesis model. (Customer location data input files must be obtained separately, however.) The interface package is designed to operate under either a Windows 95/98 or a Windows NT operating environment, though the use of Windows NT is strongly recommended. The HCPM cluster module requires Microsoft Access to function correctly, and the HAI modules require both Microsoft Access and Microsoft Excel.

B.5 Running HCPM Modules Using DOS Batch Files

Any combination of HCPM modules can be conveniently run in batch mode by using the command line argument "batch" (e.g by typing "cluster batch" or "feeddist batch" from a DOS prompt). These commands assume that particular file containing a list of wire centers as well as all necessary input files for these wire centers are present in the same folder as the executable file. For the cluster module, the batch list file is called "infiles.lst" while for both CLUSINTF and FEEDDIST the corresponding file is called "batch.lst". Additional command line arguments can be specified for each of these programs as described earlier in this section. Fully automated operation of all three modules is also possible from a DOS batch file. For example, a file hcpm.bat could be used to run the three HCPM modules when using the Windows 95 operating system. Such a file should contain the following instructions:

```
c:
cd \hcpm

dir /on /b *.in > infiles.lst
copy infiles.lst batch.lst
call cluster batch
pause

call clusintf batch -c -s0.36
pause

call feeddist batch -c -v -p500
pause
```

If the Windows NT operating system is used, the "pause" commands in the above file can be eliminated. This batch file assumes that the desired wirecenter input files (*.in) and all of the parameter input files (*.txt and *.prm) are present in the same folder as the programs cluster.exe, clusintf.exe and feeddist.exe.

References

Everitt, Brian S. (1993). *Cluster Analysis, Third Edition*, London: Arnold.

Gabel, D. and M. Kennet (1991). *Estimating the Cost Structure of the Local Telephone Exchange Network.* NRRI Report No. 91-16, Colunbus, OH.

Le, H. and W.W. Sharkey (1998), "The HCPM/HAI Interface for a Cost Proxy Model Synthesis: A User Manual."

Love, R.F., J.G. Morris and G.O. Wesolowsky, (1988), "Mathematical Models of Travel Distances," in *Facilities Location: Models and Methods.* Amsterdam: North Holland.

Gower, J.C. and G.J.S. Ross (1969), "Minimum Spanning Trees and Single Linkage Cluster Analysis," *Applied Statistics, 18,* 54-64.

Prim, R.C. (1957), "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal, 36,* 1389-1401.

Sharkey, W.W. (1995), "Network Models in Economics," Chapter 9 in M.O. Ball et al., Eds, *Network Routing: Handbooks in OR and MS, Vol. 8.* Amsterdam: North Holland.